MSC GEODETIC ENGINEERING

MSR-02: ADVANCED TECHNIQUES FOR MOBILE SENSING AND ROBOTICS (GEODESY TRACK)

02: TRAJECTORY ESTIMATION

DR. LASSE KLINGBEIL

INSTITUTE FOR GEODESY AND GEOINFORMATION UNIVERSITY OF BONN
ADVANCED TECHNIQUES FOR MOBILE SENSING AND ROBOTICS – LECTURE CONTENT

(1) Mobile Laser Scanning

(2) Trajectory Estimation

(3) System Calibration

(4) Sensor Synchronisation

(5) From Images to Point Clouds (SfM)

(6) Accuracy of Point Clouds I

(7) Accuracy of Point Clouds II

(8) Deformation Analysis with Point Clouds I

(9) Deformation Analysis with Point Clouds II
MOBILE MAPPING

Environment

Moving Platform with Sensors

Trajectory

Spatial Information about the Environment

\[ T_s^g(t) \]
CHAPTER 1: MOBILE LASER SCANNING

\[ \mathbf{p}_{\text{object}}^{\text{global}}(t_s) = \mathbf{T}_{\text{body}}^{\text{global}}(t_s) \cdot \mathbf{T}_{\text{sensor}}^{\text{body}} \cdot \mathbf{p}_{\text{object}}^{\text{sensor}}(t_s) \]

- Review of involved coordinate systems /frames
- Derivation of detailed georeferencing equation for the example of mobile laser scanning

\[
\begin{bmatrix}
x_e \\
y_e \\
z_e
\end{bmatrix} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \mathbf{R}_n^e(L, B) \mathbf{R}_b^n(\phi, \theta, \psi) \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} + \mathbf{R}_s^b(\alpha, \beta, \gamma) \cdot \begin{bmatrix} 0 \\ d \cdot \sin b \\ d \cdot \cos b \end{bmatrix}
\]
CHAPTER 2: TRAJECTORY PARAMETERS

\[
\begin{bmatrix}
x_e \\
y_e \\
z_e 
\end{bmatrix}
= \begin{bmatrix}
t_x \\
t_y \\
t_z 
\end{bmatrix}
+ \mathbf{R}_n^e (L, B) \mathbf{R}_b^n (\phi, \theta, \psi) \cdot \begin{bmatrix}
\Delta x \\
\Delta y \\
\Delta z 
\end{bmatrix}
+ \mathbf{R}_s^b (\alpha, \beta, \gamma) \cdot \begin{bmatrix}
0 \\
d \cdot \sin b \\
d \cdot \cos b
\end{bmatrix}
\]
SENSORS OVERVIEW

Navigation sensors

- Inertial sensors
- Odometer
- GNSS
- Compass
- Inclinometer

Mapping sensors

- Laserscanner
- Camera
- Radar
- Control points/Landmarks
- Bundle adjustment
- Scan Matching
- Visual Odometrie

relativ

absolut

direct

indirect
COORDINATE SYSTEMS

Inertial frame
Axis fixed with respect to the 'rest of the universe'
INERTIAL SENSORS

ACCELEROMETER

• Measurement of specific force (= non gravitational acceleration)

Specific force of the body frame relativ to the inertial frame, given in body frame coordinates

\[ f_{ib} = a_{ib} - \gamma_{ib} \]

\[ a = 9.81 \text{ \text{m/s}^2} \]

\[ g = -9.81 \text{ \text{m/s}^2} \]
INERTIAL SENSORS

GYROSCOPE

- Measure angular rate of the body relative to the inertial frame given in body frame coordinates

\[ \omega_{ib} \]
STRAP DOWN INTEGRATION

INERTIAL FRAME EQUATIONS

- Use inertial frame as the system of resolving frame for position and orientation
- Update orientation

\[ C_b^i(+) \approx C_b^i(-) (\mathbb{1} + \Omega_{ib}^b \tau_i) \]

- Transform accelerometer measurement to inertial frame
- Correct for gravity
- Update velocity
- Update position

Integration errors
TRAJECTORY PARAMETERS

\[
\begin{bmatrix}
    x_e \\
    y_e \\
    z_e
\end{bmatrix}
= \begin{bmatrix}
    t_x \\
    t_y \\
    t_z
\end{bmatrix}
+ R_n^e (L, B) R_b^n (\phi, \theta, \psi) \cdot \begin{bmatrix}
    \Delta x \\
    \Delta y \\
    \Delta z
\end{bmatrix}
+ R_s^b (\alpha, \beta, \gamma) \cdot \begin{bmatrix}
    0 \\
    d \cdot \sin b \\
    d \cdot \cos b
\end{bmatrix}
\]
STRAPDOWN INTEGRATION

\[
\begin{pmatrix}
q^n_b \\
p^e_{eb} \\
v^n_{eb} \\
\end{pmatrix}
\left.\right|_k = f \begin{pmatrix}
q^n_b \\
p^e_{eb} \\
v^n_{eb} \\
\end{pmatrix}
\left.\right|_{k-1}
, \begin{pmatrix}
a^b_{ib} \\
\omega^b_{ib} \\
\end{pmatrix}
\left.\right|_{k-1}
\]

WANTED:

Angular rate

IMU

acceleration

Earth rate

Transport rate

position

velocity

attitude

Gravity

Coriolis correction
INITIALIZATION AND ALIGNEMENT

HOW TO GET THE STARTING VALUES?

• Position
  - Use external systems, such as GNSS
  - Start from a known position (e.g. car has parked at known position)

• Velocity
  - Use external system (e.g. GNSS)
  - Start in a static situation (v=0)

• Orientation
  - Accelerometer levelling
  - Gyro Compassing
  - Magnetometer/Compass
  - Multiple GNSS Antennas
INS ERROR PROPAGATION

Orientation update

\[ C_b^i(+) \approx C_b^i(-) (1 + \Omega_{ib}^b \tau_i) \]

- Constant gyroscope error leads to orientation error

\[ \delta \phi_n^b(t) = b_\omega t \]

- e.g. 0.1 ° gyro bias \( \Rightarrow \) 30° orientation error after 5 minutes
INS ERROR PROPAGATION

Position/velocity update

\[ v_{ib}^i (+) \approx v_{ib}^i (-) + a_{ib}^i \tau_i \]
\[ r_{ib}^i (+) \approx r_{ib}^i (-) + v_{ib}^i (-) \tau_i \]

• Constant velocity error integrates to position error

\[ \delta r_{eb}^n (t) \approx \delta v_{eb}^n t \]

○ e.g. 0.1 m/s velocity error → 30m position error after 5 minutes

• Constant accelerometer bias leads to velocity and position errors

\[ \delta v_{eb}^n (t) \approx C_b^n b_a t, \quad \delta r_{eb}^n (t) \approx \frac{1}{2} C_b^n b_a t^2 \]

○ e.g. 0.01 m/s² accelerometer bias → 450m position error after 5 minutes
INS ERROR PROPAGATION

• Constant attitude errors lead to wrong transformations of the measured specific-force into the resoving frame

  o „Gravity compensation“ is wrong, parts of the measured reaction to gravity are considered as acceleration leading to acceleration offsets

  o e.g. 0.057° attitude error $\Rightarrow$ 440m position error after 5 minutes

• Constant gyroscope bias leads to attitude error leads to position error

  $\delta r_{eb}^n(t) \propto t^3$

  o e.g. 2.1°/hr bias $\Rightarrow$ 440m position error after 5 minutes
Fig. 10.
Short-term straight-line position error growth per axis for different error sources. (From [3] © Paul Groves 2013. Reproduced with permission.)
INS ERROR PROPAGATION

SENSOR NOISE

• Noisy measurements of accelerometers and gyroscopes also lead to noisy position errors (uncertainty)

---

**Fig. 12.**
Short-term straight-line position error standard deviation growth per axis because of inertial sensor noise. (From [3] © Paul Groves 2013. Reproduced with permission.)
NON-LINEAR FILTERING

• Linear Kalman Filter:
  o linear system and measurement models
    \[ x_k = \Phi x_{k-1} + Bu_k + Gw_{s,k} \]
    \[ z_k = H_k x_k + w_{m,k} \]

• General case:
  o predicted state is a function of last state, control input and noise
    \[ x_k = f(x_{k-1}, u_k, w_{s,k}) \]
  o Measurement is a function of the state
    \[ z_k = h(x_k, w_{m,k}) \]
KALMAN FILTER

Prediction

\[
x_k^- = \Phi x_{k-1}^-
\]

\[
P_k^- = \Phi P_{k-1}^- \Phi^T + Q
\]

Kalman Gain

\[
K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}
\]

\[
x_k = x_k^- + K_k (z_k - H x_k^-)
\]

Correction

\[
P_k = (I - K_k H) P_k^-
\]

System noise

Measurement noise

Predicted estimate

Updated estimate

Predicted covariance

Updated covariance

Measurement
NON LINEAR EXAMPLE (DEAD RECKONING)

SYSTEM MODEL

\[
\mathbf{x}_k = \begin{pmatrix} x_k \\ y_k \\ \phi_k \end{pmatrix} = f \left( \begin{pmatrix} x_{k-1} \\ y_{k-1} \\ \phi_{k-1} \end{pmatrix}, \begin{pmatrix} \nu_{odo,k} \\ \omega_{gyro,k} \end{pmatrix} \right) = \begin{pmatrix} x_{k-1} + \cos(\phi_{k-1}) \nu_{odo,k} \Delta t \\ y_{k-1} + \sin(\phi_{k-1}) \nu_{odo,k} \Delta t \\ \phi_{k-1} + \omega_{gyro,k} \Delta t \end{pmatrix}
\]

control input

Non linear!

Starting point \((x_0, y_0)\)
NON LINEAR EXAMPLE (DEAD RECKONING)

MEASUREMENT MODEL (E.G. GPS, DISTANCE TO KNOWN POSITION)

\[ z_k = d_k = h(x_k) = \sqrt{(x_k - x_{sat,k})^2 + (y_k - y_{sat,k})^2} \]

Starting point \((x_0, y_0)\)

Non linear!
SUMMARY EXTENDED KALMAN FILTER

**Prediction**

\[
\begin{align*}
\hat{x}_k^- & = f(x_{k-1}, u_k) \\
\hat{P}_k^- & = \Phi \hat{P}_{k-1} \Phi^T + GQG^T
\end{align*}
\]

**Correction**

\[
\begin{align*}
K_k & = \hat{P}_k^- H^T (HP_k^- H^T + R)^{-1} \\
\hat{x}_k & = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-)) \\
\hat{P}_k & = (I - K_k H) \hat{P}_k^-
\end{align*}
\]

**Derivatives**

\[
\begin{align*}
\Phi & = \left. \frac{\partial f(x, u)}{\partial x} \right|_{x = \hat{x}_{k-1}, u = u_k} , \\
G & = \left. \frac{\partial f(x, u)}{\partial w} \right|_{...} \\
H_k & = \left. \frac{\partial h(x)}{\partial x} \right|_{x = \hat{x}_k^-} 
\end{align*}
\]
**2D KF EXAMPLE**

**SYSTEM MODEL**

- No assumed (and wrong) model on the temporal evolution of motion states
- Relative measurements (path increment and angle change) are modelled as control input
- System noise is represented as control noise

\[
\begin{align*}
\mathbf{x}_k &= f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k) \\
\begin{bmatrix}
\mathbf{x}_k \\
\mathbf{y}_k \\
\mathbf{\varphi}_k
\end{bmatrix} &= f\left(
\begin{bmatrix}
\mathbf{x}_{k-1} \\
\mathbf{y}_{k-1} \\
\mathbf{\varphi}_{k-1}
\end{bmatrix},
\begin{bmatrix}
\Delta s_k \\
\omega_k
\end{bmatrix},
\begin{bmatrix}
w_{odo} \\
w_{gyro}
\end{bmatrix}
\right) =
\begin{bmatrix}
x_{k-1} + \cos(\varphi_{k-1})\Delta s_k + w_{odo} \\
y_{k-1} + \sin(\varphi_{k-1})\Delta s_k + w_{odo} \\
\varphi_{k-1} + \left(w_{gyro, k} + w_{gyro}\right)\Delta t
\end{bmatrix}
\end{align*}
\]
2D KF EXAMPLE

TRANSITION MATRIX

\[ \Phi_{k-1} = \frac{\partial}{\partial x} \begin{bmatrix} x_{k-1} + \cos(\varphi_{k-1})(\Delta s_k + w_{odo}) \\ y_{k-1} + \sin(\varphi_{k-1})(\Delta s_k + w_{odo}) \\ \varphi_{k-1} + (\omega_{gyro,k} + w_{gyro})\Delta t \end{bmatrix} \]

\[ x = \hat{x}_{k-1}, u = u_k \]

\[ \begin{bmatrix} 1 & 0 & -\sin(\varphi_{k-1})\Delta s_k \\ 0 & 1 & \cos(\varphi_{k-1})\Delta s_k \\ 0 & 0 & 1 \end{bmatrix} \]

\[ x = \hat{x}_{k-1}, u = u_k \]
2D KF EXAMPLE

\[ \begin{align*}
x_k^- &= f(x_{k-1}, u_k) \\
P_k^- &= \Phi P_{k-1} \Phi^T + GQG^T
\end{align*} \]

SYSTEM NOISE

\[ Q = \begin{bmatrix} \sigma_{odo}^2 & 0 \\ 0 & \sigma_{gyro}^2 \end{bmatrix} \]

\[ G = \frac{\partial f(x, u, w)}{\partial w} = \begin{bmatrix} \cos(\varphi_{k-1}) & 0 \\ \sin(\varphi_{k-1}) & 0 \\ 0 & \Delta t \end{bmatrix} \]

\[ x = \hat{x}_{k-1}, u = u_k \]

\[ GQG^T = \begin{bmatrix} \cos^2(\varphi_{k-1})\sigma_{odo}^2 & \cos(\varphi_{k-1}) \sin(\varphi_{k-1})\sigma_{odo}^2 & 0 \\ \cos(\varphi_{k-1}) \sin(\varphi_{k-1})\sigma_{odo}^2 & \sin^2(\varphi_{k-1})\sigma_{odo}^2 & 0 \\ 0 & 0 & \sigma_{gyro}^2 \Delta t^2 \end{bmatrix} \]
2D KF EXAMPLE

MEASUREMENT MODEL

\[
\begin{align*}
K_k &= P_k^- H^T (HP_k^- H^T + R)^{-1} \\
x_k &= x_k^- + K_k (z_k - \hat{h}(x_k^-)) \\
P_k &= (I - K_k H) P_k^-
\end{align*}
\]

- Measurement is only GPS Position \( x_{gps}, y_{gps} \)

\[
\begin{align*}
z_k &= \begin{bmatrix} x_{gps,k} \\ y_{gps,k} \end{bmatrix} = h(x_k) = H x_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x_k
\end{align*}
\]

- Measurement Noise

\[
R = \begin{bmatrix} \sigma_{gps,x}^2 & 0 \\ 0 & \sigma_{gps,y}^2 \end{bmatrix}
\]
GPS/IMU INTEGRATION

• Estimate the position and orientation of the system in all dimensions

• Use inertial sensors (acc + gyro) as control input: Strapdown inertial integration as system model

• Because acc + gyro have time dependant systematic errors (bias), estimate the biases as part of the state vector

• Use GPS as measurements
  
  o Use of positions:  ➔ loosely coupled integration

  o Use of GPS raw data (double differences, carrier phases)  ➔ tightly-coupled integration
GPS/IMU INTEGRATION

• State Vector

\[ \mathbf{x} = \begin{bmatrix} q_n^b \\ x_{eb}^e \\ v_{eb}^n \\ b_{gyro} \\ b_{acc} \end{bmatrix} \]

- Orientation quaternion
- IMU bias values

• Control input

\[ \mathbf{u} = \begin{bmatrix} f_{ib}^b \\ \omega_{ib}^b \end{bmatrix} \]

- IMU sensor readings

• Measurement

\[ \mathbf{z} = \mathbf{x}_{gps} \]

- Processed GNSS position
System noise is IMU noise
GPS/IMU INTEGRATION

MEASUREMENT MODEL

\[
K_k = P_k^{-1} H^T (H P_k^{-1} H^T + R)^{-1}
\]

\[
x_k = x_k^- + K_k (z_k - h(x_k^-))
\]

\[
P_k = (I - K_k H) P_k^-
\]

• Measurement is only GPS Position

\[
z_k = \begin{bmatrix} x_{gps,k} \\ y_{gps,k} \\ z_{gps,k} \end{bmatrix} = h(x_k) = H x_k = \begin{bmatrix} \ldots & 1 & 0 & 0 & \ldots \\ \ldots & 0 & 1 & 0 & \ldots \\ \ldots & 0 & 0 & 1 & \ldots \end{bmatrix} x_k
\]

• Measurement Noise

\[
R = \begin{bmatrix} \sigma_{gps,x}^2 & 0 & 0 \\ 0 & \sigma_{gps,y}^2 & 0 \\ 0 & 0 & \sigma_{gps,z}^2 \end{bmatrix}
\]

\[
x = \begin{bmatrix} q^n_b \\ x_{eb}^e \\ v^n_{eb} \\ b_{gyro} \\ b_{acc} \end{bmatrix}
\]
GPS/IMU INTEGRATION

GPS NOISE

Carrierphase
2-Frequency
Differential
Kinematic
Post-Processing

Remember
GPS noise is not white!

\[
R = \begin{bmatrix}
\sigma^2_{gps,x} & 0 & 0 \\
0 & \sigma^2_{gps,y} & 0 \\
0 & 0 & \sigma^2_{gps,z}
\end{bmatrix}
\]

Wrong
FILTERING/SMOOTHING

Prediction:

Filtering:

Smoothing:

Measurements

Estimate
SMOOTHING

Fixed-Point

Fixed-Lag

Fixed-Interval

Fixed Items
Varying Items
**KALMAN FILTER**

### Prediction

\[
\begin{align*}
\mathbf{x}_k^- &= \Phi \mathbf{x}_{k-1} \\
\mathbf{P}_k^- &= \Phi \mathbf{P}_{k-1} \Phi^T + \mathbf{Q}
\end{align*}
\]

### Kalman Gain

\[
K_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1}
\]

### Correction

\[
\begin{align*}
\mathbf{x}_k &= \mathbf{x}_k^- + K_k (\mathbf{z}_k - \mathbf{H} \mathbf{x}_k^-) \\
\mathbf{P}_k &= (\mathbf{I} - K_k \mathbf{H}) \mathbf{P}_k^-
\end{align*}
\]
FORWARD BACKWARD FILTER

\[ \hat{x}_u = k_{F_k} \hat{x}_{F_k} + k_{B_k} \hat{x}_{B_k} \]

\[ = k_F x_f + (1-k_F)x_0 \]

\[ \rho(x) = \varepsilon \left[ (x-x)(x-x)^T \right] \]

minimize \( \rho \)

\[ \Rightarrow k_F = \frac{\rho_0}{\rho_F + \rho_0} \]

\[ k_B = \frac{\rho_F}{\rho_F + \rho_0} \]

\[ \rho = \left( \rho_F^{-1} + \rho_0^{-1} \right)^{-1} \]

weighted mean
FORWARD BACKWARD FILTER

\[
\begin{align*}
\mathbf{x}_{S,k} &= K_{F,k} \mathbf{x}_{F,k} + K_{B,k} \mathbf{x}_{B,k}^-
onumber \\
&= K_{F,k} \mathbf{x}_{F,k} + (1 - K_{F,k}) \mathbf{x}_{B,k}^- \\
K_{F,k} &= P_{R,k}^{-1}(P_{F,k} + P_{B,k}^-)^{-1} \\
P_{S,k} &= \left[ (P_{F,k})^{-1} + (P_{B,k}^-)^{-1} \right]^{-1}
\end{align*}
\]

Details on the derivation in:
Dan Simon: Optimal State Estimation, Wiley Interscience
RAUCH-TUNG-STRIEBEL (RTS) SMOOTHER

1) \( \hat{x}_f, \hat{p}_f \)

2) \( o \rightarrow t \)

\[ \hat{x}_k = \hat{x}_f_k + \Phi_k \left( \hat{x}_{k+1} - \hat{x}_{f_{k+1}} \right) \]

\[ A = \frac{\hat{p}_f_k \cdot \Phi_k^T}{\hat{p}^-_{f_{k+1}}} \]
RAUCH-TUNG-STRIEBEL (RTS) SMOOTHER

Runs from $k$ to $1$

\[
x_{S,k} = x_{F,k} + A_k(x_{S,k+1} - x_{F,k+1})
\]

\[
A_k = P_{F,k} \Phi_k^T \left[P_{F,k+1}^-\right]^{-1}
\]

\[
P_{S,k} = P_{F,k} + A_k(P_{S,k+1} - P_{F,k+1}^-)
\]

Details on the derivation in:
Dan Simon: Optimal State Estimation, Wiley Interscience
## HOW TO EVALUATE THE QUALITY OF THE ESTIMATION?

### A) LOOK AT THE COVARIANCE OF THE ESTIMATE

**Prediction**

\[
\begin{align*}
x_k^- &= \Phi x_{k-1} \\
P_k^- &= \Phi P_{k-1} \Phi^T + Q
\end{align*}
\]

**Correction**

\[
\begin{align*}
K_k &= P_k^- H^T (HP_k^- H^T + R)^{-1} \\
x_k &= x_k^- + K_k (z_k - H x_k^-) \\
P_k &= (I - K_k H) P_k^-
\end{align*}
\]

- Does not depend on measurements
- Only 'valid', if all models and uncertainties are 'correct'
- No systematic errors included
HOW TO EVALUATE THE QUALITY OF THE ESTIMATION?

B) LOOK AT THE INNOVATION

\[
\begin{align*}
x_k^- &= \Phi x_{k-1} \\
P_k^- &= \Phi P_{k-1} \Phi^T + Q
\end{align*}
\]

Prediction

\[
\begin{align*}
K_k &= P_k^- H^T (H P_k^- H^T + R)^{-1} \\
x_k &= x_k^- + K_k (z_k - H x_k^-) \\
P_k &= (I - K_k H) P_k^-
\end{align*}
\]

Correction

- Gives an idea about the consistency between models and observations
- It is not clear if inconsistencies come from wrong measurements or predictions
HOW TO EVALUATE THE QUALITY OF THE ESTIMATION?

C) CAREFULLY LOOK AT ALL ESTIMATED STATES

- A nice looking position result can also appear if your process model is completely wrong and your measurements are good
- E.g. Estimated bias values show ‘consistency’ of models
HOW TO EVALUATE THE QUALITY OF THE ESTIMATION?

D) COMPARE WITH REFERENCE SOLUTION

- Needs synchronization between reference and estimation results
- Difficult to realize in outdoor scenarios (especially for orientation)
- How to 'parametrize' the error? Depends on motion, environment, time ...
HOW TO EVALUATE THE QUALITY OF THE ESTIMATION?

REFERENCE SYSTEMS

- Total station
- Tracking systems
- Reference trajectory
- RTK GNSS
HOW TO EVALUATE THE QUALITY OF THE ESTIMATION?

E) INDIRECT EVALUATION

\[
\begin{bmatrix}
x_e \\
y_e \\
z_e
\end{bmatrix} = \begin{bmatrix}
t_x \\
t_y \\
t_z
\end{bmatrix} + \mathbf{R}_n^e (L, B) \mathbf{R}_b^n (\phi, \theta, \psi) \cdot \begin{bmatrix}
\Delta x \\
\Delta y \\
\Delta z
\end{bmatrix} + \mathbf{R}_s^b (\alpha, \beta, \gamma) \cdot \begin{bmatrix}
0 \\
d \cdot \sin b \\
d \cdot \cos b
\end{bmatrix}
\]

- Includes many other errors
- Assignement of error to specific sources is difficult
EXAMPLE
track

LC ESKF
Novatel GNSS
KF WITH ODOMETER. NO GPS!
GNSS QUALITY

Number of GNSS satellites

- Minimum number
- Used number

Standard deviation

- $\sim \text{cm}$
- $5 \times 10^{-8}$ m
Track comparison: RTS smoother without and with GNSS outages
DIFFERENCE TO GNSS DATA

East

Track time [s]

North

Height

Track time [s]
ZOOM 2

The image shows a track plot with different markers indicating various methods and conditions:

- **LC ESKF**
- **RTS Smoother**
- **Novatel GNSS**
- **GNSS outage**

The plot is labeled with East and North axes, and the track is marked with a series of dots and lines indicating the trajectory over a geographic area.
BIAS ESTIMATION

Accelerometer bias

Gyroscope bias

Klingbeil: Advanced Techniques for Mobile Sensing and Robotics - Geodesy - 02 - Trajectory Estimation
WHAT YOU HAVE LEARNED TODAY

• How are position and orientation calculated using inertial sensors and GNSS?

• How are inertial sensors and GNSS observations combined using filtering algorithms?

• How are inertial sensor errors influencing the estimation?

• What is the difference between filtering and smoothing and when/why is smoothing better?

• What is the RTS smoother?

• How can you evaluate the quality of a trajectory estimation algorithm/system?
THANKS